

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

**Návrh portálu
"Personalistika a mzdy"
na podnikovém intranetu
ArcelorMittal**

**Design of a Portal
"Human Resources and Payroll"
in the Enterprise Intranet of ArcelorMittal**

Zadání bakalářské práce

Student:

Petr Piskorz

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Návrh portálu "Personalistika a mzdy" na podnikovém intranetu
ArcelorMittal
Design of a Portal "Human Resources and Payroll" in the Enterprise
Intranet of ArcelorMittal

Zásady pro vypracování:

Cílem práce bude navrhnout nové řešení portálu Personalistiky a mezd na podnikovém intranetu. Současný stav je nepřehledný a neodpovídá grafickým šablonám podle norem ArcelorMittal.

Zásady pro vypracování:

1. Proved'te analýzu současného stavu a nastuduje předepsané šablony dodané korporátem.
2. Vytvořte návrh nového portálu. Portál nabídne uživatelům zadávat a tisknout různé formuláře a sestavy, vyhledávat informace v katalozích a být upozorňován na novinky v mzdovém a personálním systému. Toto bude realizováno pomocí již hotových aplikací.
3. Postupně migrujte web aplikace z prostředí IBM DB2 NetData. Migrovány budou aplikace pro tvorbu excelovských sestav odboru Personalistika. Podle odhadu se bude jednat o 3-4 aplikace.
4. Návrh presentační vrstvy proved'te v prostředí Websphere Application Developer 5.0, což je IBM nástavba Elipse, která je používána pro vývoj WEB aplikací v podniku ArcelorMittal.
5. Pro výběr dat použijte již hotové vložené procedury DB2.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Petr Kožušník**

Konzultant bakalářské práce: Ing. Jan Kožusznik, Ph.D.

Datum zadání: 19.11.2010

Datum odevzdání: 04.05.2012



Eduard Sojka

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

Gm

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 4. května 2012

Peslva
.....

Rád bych na tomto místě poděkoval všem, kteří mi pomohli, protože bez nich by tato práce nevznikla. Především děkuji vedoucímu bakalářské práce panu Ing. Petru Kožušníkově a konzultantovi Ing. Janu Kožuszníkovi, Ph.D. Díky také patří моým rodičům za podporu.

Abstrakt

Cílem práce je navrhnout nové řešení portálu Personalistiky a mezd na podnikovém intranetu. Současný stav je nepřehledný a neodpovídá grafickým šablonám podle norem ArcelorMittal. Bylo zapotřebí naprogramování několika JavaScriptových aplikací a funkcí pro zefektivnění ovládání uživatelského rozhraní.

Klíčová slova: portál, grafické rozhraní, IBM DB2 NetData, Websphere Studio Application Developer V.5.0

Abstract

The aim of the thesis is designing a new portal solution for human resources and payrolls in company's intranet. The current situation is confusing and graphic templates do not correspond with standards according to ArcelorMittal. I had to program some JavaScript applications and functions to make a user interface more efficient.

Keywords: portal, graphical interface, IBM DB2 NetData, Websphere Studio Application Developer V.5.0

Seznam použitých zkratk a symbolů:

HTML	- HyperText Markup Language (značkovací jazyk pro webové stránky)
CSS	- Cascading Style Sheets (kaskádové styly)
JSP	- Java Server Page
JS	- JavaScript
PC	- Personal computer (osobní počítač)
IBM	- International Business Machines Corporation
Java EE	- Java Enterprise Edition
EAR	- Enterprise Archive
TSO	- Time Sharing Option
RGB	- Red Green Blue (způsob míchání barev)
WSAD	- Websphere Studio Application Developer
SQLJ	- Structured Query Language for Java

Obsah

1. Úvod.....	3
2. Návrh portálu	4
3. Základní popis portálu.....	5
4. Kategorie portálu.....	6
4.1 Obecné.....	6
4.2 Datové centrum	6
4.3 Reporting.....	6
4.4 Instrukce	6
4.5 Návodý	7
4.6 Prezentace	7
4.7 Dokumentace.....	7
5. Použité technologie	8
5.1 HTML	8
5.2 JavaScript	8
5.2.1 Charakteristika JavaScriptu.....	8
5.3 JSP (Java Server Page).....	9
5.4 DB2 Stored procedure (uložené procedury).....	10
5.5 Kaskádové styly (CSS).....	10
5.6 WebSphere Application Server (WAS)	11
5.7 NET Data	11
5.8 SQLJ.....	11
6. Realizace	12
6.1 Migrace do prostředí WSAS 5.0	12
6.2 Obsahové a funkční změny	13
6.3 Grafická část.....	14
7.1 Instalace konkrétní aplikace portálu.....	15
8. Migrace NET Data do prostředí WSAD 5.0	18
8.1 Popis prostředí.....	18
8.2 Migrace	19
9. Migrace z transakčního prostředí do webového za využití uložených procedur DB2	26

9.1 Proces s využitím transakčního manažeru.....	26
9.2 Proces s využitím webových služeb.....	26
9.3 Vytvoření uložené procedury DB2	27
10. Závěr	33
11. Reference.....	34

1. Úvod

V dnešní době si už jen stěží dokážeme představit hledání veškerých informací v šanonech, případně archivech. Prakticky již nenajdeme firmu bez webových stránek. Stejně tak důležité jako jsou webové stránky pro potenciálního zákazníka, kde přijde s naší firmou poprvé do kontaktu, jsou portály pro zaměstnance. Takový portál si můžeme představit jako bránu zprostředkovanou webovým serverem pro efektivní získávání informací. Přestože největší slávu si portály získaly již v 90. letech 20. století, kdy vyhledávače neposkytovaly tak přesné výsledky jako ty dnešní, těší se i dnes u uživatelů velké oblibě. Nejvíce pak u lidí se základními znalostmi práce s PC. Můžeme si jej zjednodušeně představit jako katalog odkazů.

Portál, který vznikl v rámci bakalářské práce, byl určen pro ArcelorMittal a. s. Jedná se prakticky o největší společnost v rámci Moravskoslezského kraje zabývající se hutní výrobou. Tato společnost zaměstnává více než 5400 lidí. O tyto lidi se starají personalisté a to v každém ohledu, ať už mluvíme o jejich platech, dovolených, či čerpání různých zaměstnaneckých programů. Úkolem bylo právě personalistům ulehčit práci novým portálem.

2. Návrh portálu

Nejprve bylo zapotřebí uvědomit si, jaká cílová skupina bude s portálem pracovat. Jelikož koncoví uživatelé portálu budou bez hlubších znalostí IT, byly kladeny vysoké nároky na jednoduchost a přehlednost jeho ovládání.

Poté bylo nutné zvolit vhodnou strukturu portálu. Po několika konzultacích jsem dospěl ke konečnému rozhodnutí o tom, jakým způsobem budou data kategorizována a jak budou tvořit jednotlivá menu.

Při hrubém návrhu jsem použil dva rámy, jeden vodorovný ve vrchní části a druhý svislý vlevo, s tím, že zobrazovací okno tvoří zbytek volného prostoru. Bylo nutné zachovat firemní politiku a dodržet barevnou paletu. Z mého pohledu by bylo mnohem jednodušší pro tyto účely použít PHP, už jen kvůli zásadnímu rozdílu, že Java je kompilovaný a PHP interpretovaný jazyk, případně snazší práci s řetězci. Na druhou stranu Java je podstatně rychlejší.

3. Základní popis portálu

V první řadě se jedná o portálu pro konkrétní oddělení. Jednotlivé části portálu jsou sdruženy v jeden celek pod hlavní částí, kterou si lze představit jako rozcestník. Tento rozcestník je realizován pomocí SharePoint 2010 využívající Windows Server 2008 a databázi SQL. Jedná se o prostředí, v němž se uživatelé dovědí o novinkách, které mohou přímo ovlivnit jejich práci, např. nové požadavky vedení nebo třeba jen zjednodušení práce použitím již vytvořených webových aplikací pro získání sestav.

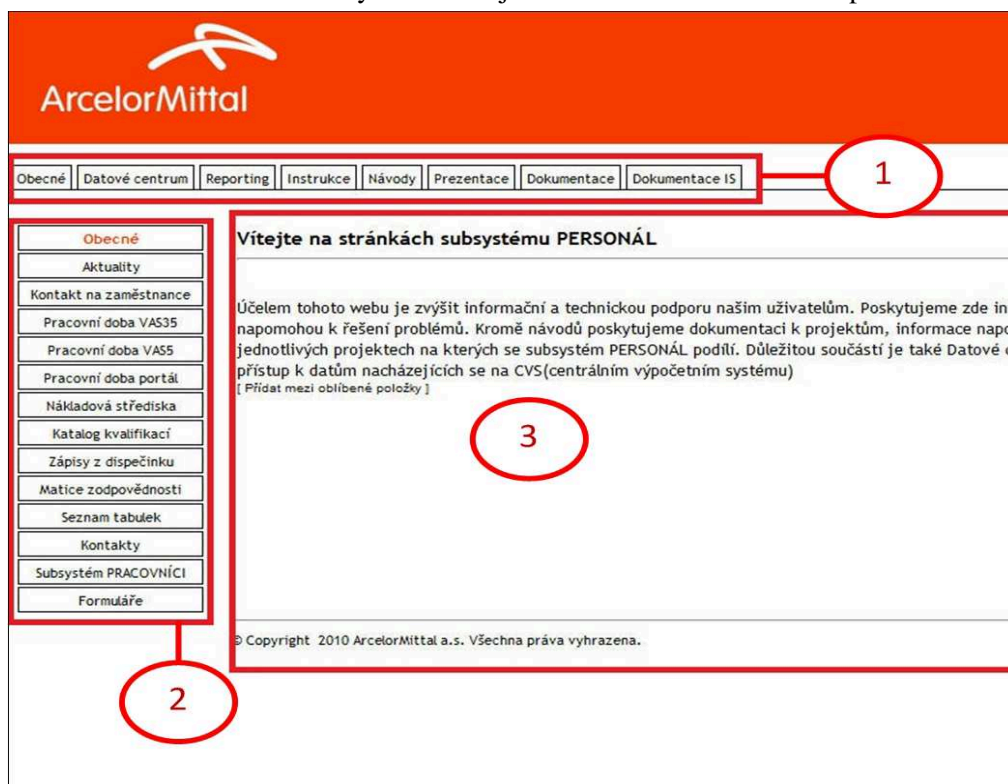
V původní podobě se v tomto prostředí uživatelé hůře orientovali. Byly zde opomenuté nabídky v menu na již neexistující oddělení, duplicita, odkazy na nefunkční cíle, kompletně jiná grafika. Většina zdrojového kódu byla generovaná, přičemž editace takto získaného kódu je velice obtížná.

Uživatelé pracují v okně, které je rozděleno na tři části pomocí rámců (frames). Tyto části můžeme vidět na obrázku 1.

První část obsahuje hlavní nabídku, jež dělí další možnosti právě podle cílové skupiny uživatele. Lze říci, že v této nabídce se určí konkrétní část portálu, se kterou se bude dále pracovat. Tato část je znázorněna na obrázku č. 1 pod číslem 1.

Druhá část obsahuje v závislosti na předchozím výběru další možnosti pro zúžení požadovaného výsledku. Můžeme tedy říci, že se vytváří kategorie. Tato část je znázorněna na obrázku č. 1 pod číslem 2.

Třetí část, hlavní okno portálu, nám zobrazí již konkrétní výsledek, případně výsledky s níž uživatel pracuje. Tímto způsobem je logicky seskupeno více než 200 odkazů na dokumenty nebo konkrétní webové stránky. Tato část je znázorněna na obrázku č. 1 pod číslem 3.



Obrázek 1: Vzhled portálu

4. Kategorie portálu

4.1 Obecné

Kategorie „Obecné“ obsahuje nejčastěji používané položky portálu, např. aktuality, které jsou zdrojem nových informací pro všechny uživatele portálu, kontakt na zaměstnance, jenž zobrazí data ze serveru, formuláře nebo pracovní doby pro konkrétní pracoviště. Dále také vstup k evidenci pracovní doby nebo katalog pro vzdělávání.

4.2 Datové centrum

Datové centrum se skládá z aplikací, vytvářejících dokumenty určené k tisku, přehledné tabulky dat, data, která lze uložit pomocí souboru na lokální disk, nebo soubory ve formátu MS Excel, jež mohou být dále zpracovány uživatelem. Tato oblast portálu je zaměřena pro již konkrétní oddělení. Tyto oddělení jsou tři:

- PaM – oddělení práce a mzdy, obsahující odkazy na sestavy umožňující vytváření dokumentu s využitím osobních dat a tvorba jednotlivých sestav s možností exportu do Excelu
- Personalisté - personální data všech zaměstnanců
- Mzdové účetní přehledy mzdových sestav a tisk potvrzení o mzdovém příjmu
- Uchazeči - tvorba dokumentů a přehledu pro nábor nových zaměstnanců
- Bezpečnost - elektronická evidence úrazů a statistické přehledy

V dalších částech menu jsou data společná pro všechny oddělení.

4.3 Reporting

Je určen především pro ředitele závodů a umožňuje přehled konkrétních ekonomicky-mzdových informací, absencí zaměstnanců a je zde také možnost srovnání s předchozím obdobím (až 4roky zpět).

4.4 Instrukce

Zde nalezneme komplexní popis informačního systému. Obsahuje výčet on-line formátů, vazby mezi položkami a pracovní postupy. Informace jsou vždy ve formátu HTML, který je zobrazitelný v prohlížeči, a ve formě MS Word. Instrukce jsou zpracovány v jednom dokumentu za všechny odborné útvary a také pro každý jednotlivý útvar.

4.5 Návody

Tato kategorie obsahuje návody k nastavení prostředí, zahrnuje jednotlivé pracovní postupy, eventuálně dokumenty popisující nastavení uživatelského prostředí např. nastavení webového prohlížeče, Accessu pro datové sklady či nastavení tiskárny pro tisk dokumentů.

4.6 Prezentace

V prezentacích lze najít souhrnné informace k novým projektům, přehledy jednotlivých projektů a komplexní přehled informačního systému PERSONÁL.

4.7 Dokumentace

V dokumentaci lze nalézt soubory ve formátu MS Word a MS Excel, které dokumentují jednotlivé části a změny informačního systému.

4.8 Dokumentace IS

Tato dokumentace je určena pouze pro zaměstnance IT sektoru a správce subsystému PERSONÁL. Obsahuje pomocné soubory pro vývojáře portálu.

5. Použité technologie

5.1 HTML

HTML, nebo-li HyperText Markup Language, je značkovací jazyk, který má velice přesnou syntaxi. Těmto značkám se říká tagy. Je exaktně definován, přitom je však velice flexibilní. Uvádí se, že existuje přes 90% webových stránek nesplňujících přesně pravidla HTML, přesto se ale zobrazí. Všechny tagy jsou uzavřené ostrými závorkami například `<body>`, a jsou rozděleny na dvě hlavní skupiny: párové a nepárové.

Rozdíl mezi nimi je zřejmý již z názvu. Párové tagy obsahují začáteční a koncový tag, nepárové nikoli.

Typickým příkladem párových tagu je nadpis první úrovně `<h1></h1>`. Mezi těmito tagy je text, který chceme stylizovat na úroveň h1.

5.2 JavaScript

JavaScript, původním názvem LiveScript, je programovací jazyk vytvořený společností Netscape. Za jeho činnost zodpovídá prohlížeč webových stránek uživatele. JavaScript tedy není vázán na server, tak jako stránky na něm uložené. Při načtení stránky dojde k provedení kódu. Jedná se tedy o jazyk, který je interpretován na straně klienta. JavaScript je velice rozšířený a je to možnost jak jednoduše a efektivně zpestřit stránky a v omezené míře ovlivnit interaktivitu webu (validace formulářů, animace objektů).

Bohužel i tento nástroj má svou stinnou stránku, tou je především rozdílná interpretace v různých prohlížečích. Proto v dnešní době pokračuje snaha JavaScript standardizovat, což ocení hlavně vývojáři vzhledem k jejich ušetřené práci. Možným řešením je používání JS knihoven a frameworků typu JQuery obsahující optimalizované funkce a objekty.

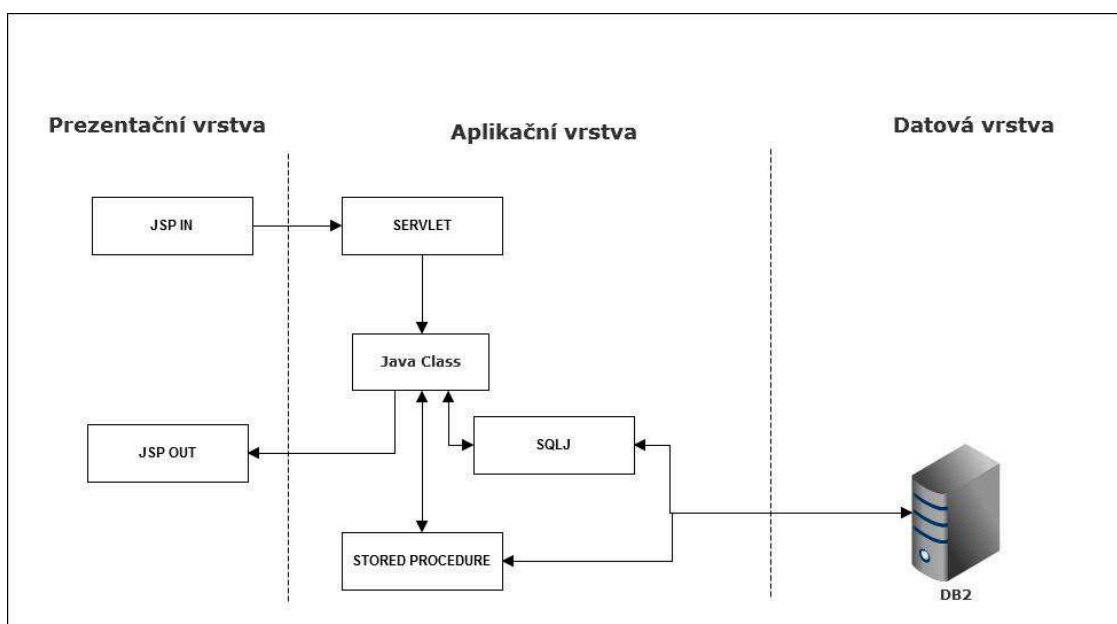
5.2.1 Charakteristika JavaScriptu

- Case-Sensitive
- Skriptovací jazyk na straně klienta
- Multiplatformní
- Dynamická typovost
- Přepis obsahu, atributů a kaskádových stylů jednotlivým HTML elementům
- Podpora JSON

5.3 JSP (Java Server Page)

Jedná se o technologii, která pomáhá vývojářům k vytvoření dynamicky generovaných webových stránek typu HTML, XML atp. Je na první pohled podobná technologii PHP. Instrukce se zapisují mezi značky `<? a ?>`. JSP jsou překládány na servlety při spuštění. Každý JSP servlet je uložen v cache paměti a je používán, dokud není původní JSP změněno. Pro naši práci jsem využil tří vrstvou architekturu.

V mém případě prezentační část zajišťuje použití právě JSP, přičemž JSP IN vstup odešle na servlet, ke kterému je vázaná Java třída. Ta je schopná pracovat jak s SQLJ, tak uloženými procedurami. Rozdíl v použití SQLJ a uloženými procedurami je ten, že uložené procedury použijeme při získávání většího množství dat (sestav). U SQLJ je to naopak. Pokud získáváme např. pouze jeden záznam, výsledek dotazu nám zpracuje opět Java třída, která vrátí výsledek formou JSP OUT.



Obrázek 2: Schéma 3vrstvé architektury JSP

5.4 DB2 Stored procedure (uložené procedury)

Uložené procedury jsou uživatelem psané programy, které jsou uloženy v databázi serveru a mohou být volány klientskou aplikací pomocí SQL. Uložená procedura může posílat a/nebo přijímat parametry z volaného programu. Hlavním cílem uložených procedur je snížení toku zpráv mezi žadatelem (aplikací) a data serverem (DB2) v distribuovaném prostředí. Dalším využitím uložených procedur je vytvoření funkce volatelné z libovolné aplikace nebo platformy. Je to jedna z nejsnazších možností jak provést vzdálené volání a distribuovat logiku aplikačního programu. Mezi hlavní výhody patří fakt, že stačí udržovat pouze jediný zdroj každé funkce, dokonce i v případě volání funkcí prováděných v odlišných prostředích. Uložené procedury se na ArcelorMittalu vyvíjí prostřednictvím Visual Age Generator 4.5

5.5 Kaskádové styly (CSS)

Jedná se o nadstavbu vyznačovacího jazyku HTML, XHTML či XML. Slouží k popisu prezentace dokumentů, aniž by ovlivnily jejich obsah. Má tedy na starosti vzhled stránky. V dnešní době se stránky rozdělují na dvě části. Jedna má na starost vzhled a druhá, ovlivňuje obsah a strukturu stránky. CSS umožňují jednoduše a rychle měnit jednotlivé parametry designu, barvy, formátování textu, pozice a mnoho dalších. Tyto styly jsou součástí stránky. Kaskádové styly můžeme zavést třemi způsoby:

- externím souborem napojeným na stránku formou reference v hlavičce HTML,
- pomocí atributu „style“ ke konkrétnímu elementu, kde jsou vyjmenovány všechny požadované parametry,
- přímo v hlavičce kódu stránky HTML pomocí tagu STYLE.

CSS jsou podobně jako JavaScripty zpracovány prohlížečem na straně klienta. Zpracovávají se postupně podle přidělených priorit. Pravidlo pro přidělování priority funguje na základě adresování. Čím je konkrétnější adresace prvku, tím vyšší prioritu tento prvek má. Tedy adresace na konkrétní element zvětší váhu více, než adresace třídy, do níž tento element spadá. Pro přesvědčení prohlížeče hrubou silou o prioritě konkrétních prvků, navzdory hierarchii dokumentu a kaskádových stylů, lze použít za parametr - klíčové slovo „!important“.

Stejně jako JavaScript má CSS rozdílnou interpretaci v různých prohlížečích. Dnes panuje viditelná snaha používat standardy, které nové prohlížeče dodržují. Nejznámější je konsorcium W3C. Jedná se o soubor standardů (doporučení), jejichž používání se stalo normou pro tvorbu webu. W3C je spravováno několika institucemi z USA, Evropy a Japonska.

5.6 WebSphere Application Server (WAS)

Co se týče softwarových produktů firmy IBM, WAS je vlajkovou lodí této firmy. První verze WAS byla k dispozici již v roce 1998. Tento produkt nám umožňuje vytvořit tzv. aplikační server s použitím otevřených standardů, jako jsou Java EE, XML a webové služby. Pracuje na všech dnes používaných platformách. Dokáže pracovat s celou škálou webových serverů, počínaje Apache HTTP Server, konče IBM HTTP Serverem pro AIX/Linux/Microsoft Windows/Solaris. Ačkoliv v dnešní době je k dispozici ve verzi 8, práce byla realizována ve WAS 5.0, který je v ArcelorMittalu provozován. V současné době se připravuje na verzi 7.0.

5.7 NET Data

Další produkt firmy IBM, pomocí kterého můžeme vytvářet dynamické webové stránky s využitím dat z relačních, i bez schematických databázových systémů řízení (např. DB2) a databáze, které mohou být přístupné prostřednictvím DRDA (Distributed Relational Database Architecture) nebo aplikací naprogramovaných prostřednictvím jazyků jako je Java, JavaScript, C, C++, COBOL a REXX. Net Data je makro procesor, jenž pracuje jako middleware na webovém serveru. Pod slovem middleware si lze představit počítačový software spojující komponenty softwaru nebo aplikace.

Net Data pracují pomocí aplikačních programů nazývaných makra. Net Data obsahuje systémové funkce, např. pro práci s řetězcí, formátování textu, matematické operace a práci s daty (kalendářními) a časem.

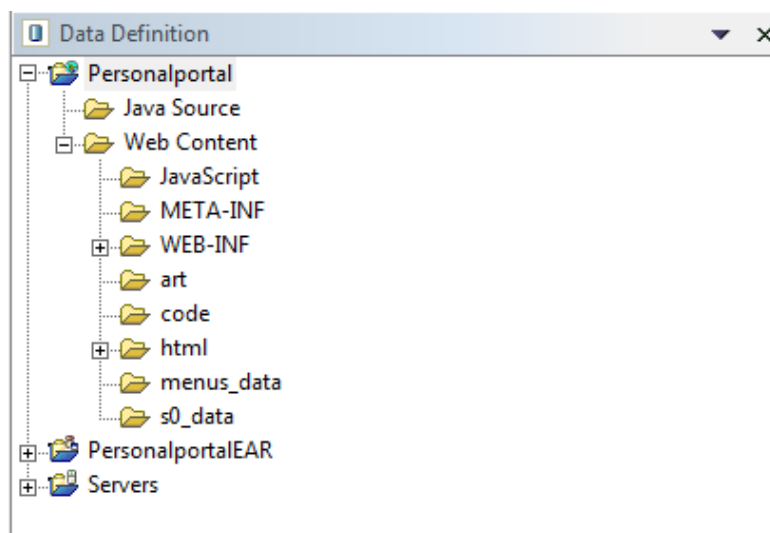
5.8 SQLJ

Jedná se o nástavbu nad JDBC (Java Database Connectivity), umožňující vkládat SQL příkazy přímo do kódu Javy. Jednou z výhod je, že kód v jazyce SQLJ nebývá tak obsáhlý jako v JDBC. Mezi další výhody můžeme jmenovat, že SQL syntaxe je ověřena při kompilaci. Samozřejmě má i své nevýhody, například postrádá podporu většiny používaných frameworků (Hibernate, atp.).

6. Realizace

6.1 Migrace do prostředí WSAS 5.0

Původní portál tvořil soubor HTML stránek, obrázků, dokumentů, stylů a skriptů tříděných pouze do jednoduchých adresářů. Prvním krokem tedy byla migrace do projektu v prostředí Websphere Studio Application Server 5.0. Toto prostředí nám umožňuje importovat celou souborovou strukturu najednou a sám si jej roztřídit podle obsahu. Viz obrázek 3.



Obrázek 3: Projekt v prostředí Websphere application server 5.0

6.2 Obsahové a funkční změny

Dalším krokem byla změna obsahu HTML stránek. Jako příklad bych uvedl změnu chování menu pomocí JavaScriptu. Bylo potřeba vytvořit takovou strukturu, aby bylo možné libovolně měnit pořadí, přidávání, případně mazání prvků v menu, aniž by se muselo měnit cokoli jiného. Nebýt tohoto požadavku, vyřešil bych problém pomocí identifikátoru jednotlivých elementů. Jednotlivé prvky menu se zobrazují s jejich podnabídkou, pokud se na ně klikne myší. Při opětovném kliknutí, případně při kliknutí na jiné menu, se podnabídka zavře a otevře se námi požadovaná.

```
<script>

    function shMenu(elem){
        var menu = document.getElementById('seznam');
        var uls = menu.getElementsByTagName('ul');
        var ul = elem.getElementsByTagName('ul')[0];
        var disabled = (ul.style.display == 'none');
        for(var i=0; i<uls.length; i++){
            uls[i].style.display = 'none';
        }
        if(disabled)
            ul.style.display = 'block';
    }

</script>
```

Výpis 1: Ukázka použití JavaScriptu

6.3 Grafická část

V okamžiku kdy jsem dokončil všechny obsahové a funkční změny v HTML a JavaScriptu, mohl jsem přistoupit ke změně grafické části a to prostřednictvím kaskádových stylů (CSS). Vzhled celého portálu je uložen v souboru *personal.css*, viz výpis 2. Změny se týkaly především barev, písma a použitých log firmy. Barvy jsou vázány na jednotlivé elementy, uvádí se anglickým názvem barvy (< font color = "red" >) nebo jedním ze zápisů pomocí RGB:

- procentuálním RGB zápisem ()
- desetinným (< font color ="rgb(255, 0, 0)">)
- šestnáctkovým (< font color ="#ff0000">)
- zkráceným šestnáctkovým () - pouze v případě kdy se všechny dvojice opakuje

V našem případě nejvýraznější barvou je #FF3700.



Další změnou bylo použité písmo, kterému se jinými slovy říká font. Firemní politikou určený font je Trebuchet MS, který vypadá takto: "Ukázka použití fontu Trebuchet MS"

```

A.banner          { color: #000000;text-decoration: none;}
A:link.banner      { color: #000000;}
A:visited.banner   { color: #000000;}

A:active.banner    { color: #000000;}

A:hover.banner     { text-decoration: underline;color: #FF3700;}
BODY.banner        {
    font-family: "Trebuchet MS";
    background-color: #FFFFFF;
    color: #FFFFFF;
    margin : 0;padding : 0; width: 100%;
    margin-top=0;
    margin-left=0;
    margin-right=0;
}
TD.ItemBody        { font-family: "Trebuchet MS";font-size : 8pt;}
TABLE.banner       { font-family: "Trebuchet MS";margin-top : 0;margin :
0;padding : 0; width: 100%;}

```

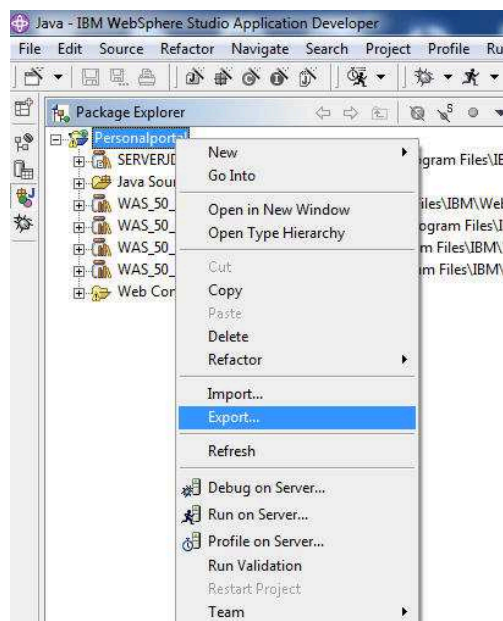
Výpis 2: Část obsahu personal.css

7.1 Instalace konkrétní aplikace portálu

Po vytvoření struktury portálu podle požadavků je potřeba nainstalovat aplikaci na samotný server. Tento proces zahrnuje tyto kroky:

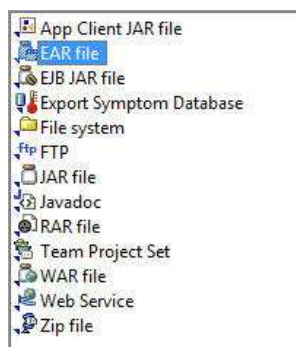
a) Export celého portálu z aplikace WAS do formátu EAR.

Jedná se o jednoduché označení projektu pravým tlačítkem myši a vybrání možnosti „Export“, viz obrázek 4.



Obrázek 4: Export z prostředí WAS 5.0

Dále je zde možnost výběru z třinácti různých formátů (obrázek 5), přičemž pro potřeby WAS je nejvhodnější použít právě EAR file.



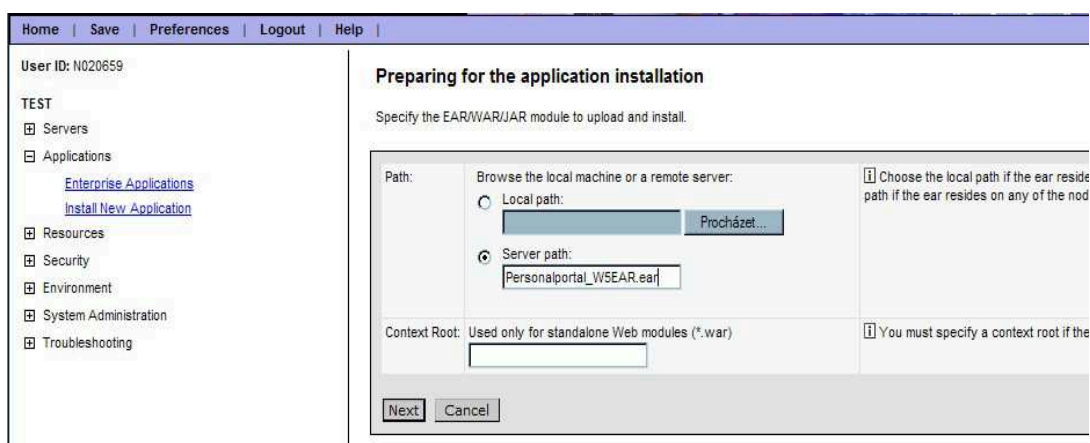
Obrázek 5: Typy souborů určených k exportu

Zde můžeme vidět, že EAR file není jediná možnost archivace, do níž můžeme svůj projekt exportovat.

EAR je souborový formát používaný Javou EE pro zabalení jednoho nebo více modulů do jediného archivu. Je to především proto, aby jednotlivé moduly byly na aplikačním serveru současně a jednotně uloženy.

b) Instalace na samotný server

Existují tři možnosti jak umístit EAR na server. První možností je nahrání EAR přes FTP na server, což můžeme realizovat například pomocí souborových manažerů, za předpokladu že je integrován FTP klient. Nejznámějším zástupcem je Total Commander. Další dvě možnosti jsou přístupné v rámci konzole již přímo na serveru. Po přihlášení do konzole v levém menu je možnost Install New Application, v hlavním okně se po vybrání této možnosti objeví všechny na serveru instalované aplikace. Po zvolení Install jsou na výběr dvě cesty: lokální cesta (tuto lze zvolit, pokud je EAR na PC, s kterým momentálně pracuje), nebo cesta k serveru, což se v praxi ukázalo jako masověji používaná možnost. Instalátor již sám vytvoří adresářovou strukturu, pomocí EAR. Tato struktura bude totožná jako při vývoji samotného portálu.

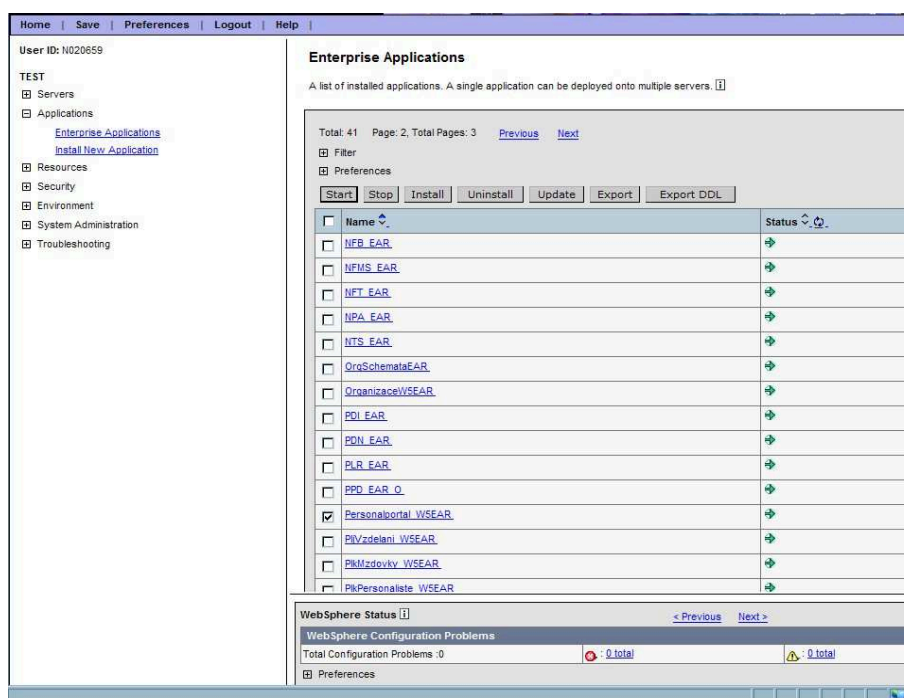


Obrázek 6: Instalace aplikace

c) Spuštění

Posledním krokem je samotné spuštění aplikace. Opět je to realizováno pomocí serverové konzole – pomocí položky Enterprise Application. Vidíme opět nainstalované aplikace již doplněné o námi vytvořenou. Aplikaci spustíme buď pomocí zaškrtnutí pole a poté tlačítkem Start, nebo přímo na řádku s názvem aplikace změnou statusu na Start pomocí kliku na jeho ikonu. Jsou zde i další možnosti jak pracovat s aplikacemi na serveru

- Start – spuštění aplikace
- Stop – zastavení spuštěné aplikace
- Install – instalace aplikace z EAR
- Uninstall – odinstalace aplikace
- Update – aktualizuje aplikaci
- Export – export aplikace (např. zpět do EAR)
- Export do DLL – export aplikace do dynamicky linkované knihovny



Obrázek 7: Spuštění aplikace

8. Migrace NET Data do prostředí WSAD 5.0

8.1 Popis prostředí

Net Data byla používána jako první prostředek pro prezentaci dat ve webovém prostředí. Do této doby byla data prezentována pouze pomocí transakčního manageru emulátoru 3270, čímž došlo k vylepšení grafických prvků a byl umožněn tisk personálních dokumentů a zobrazování generovaných sestav.

Kód obsahuje kromě volání uložených procedur také HTML kód pro zobrazení obsahu. Nevýhodou je, že makra jsou napsána v prostředí TSO. Toto prostředí nedisponuje žádnou možností exportu, kód je velice nepřehledný. Jako jednu z nevýhod bych chtěl uvést, že je schopný zobrazit pouze dvacet čtyři řádky kódu na jedné obrazovce, viz obrázek 8, přičemž samotný posun v kódu je realizován pomocí kláves F7 a F8. Net Data byla v ArcelorMittalu použita jako první technologie pro prezentaci dat na portálu. V současné době se od vývoje v tomto prostředí upouští a probíhá migrace aplikací do prostředí vývojového prostředí WSAD.

```

Relace A - [24 x 80]
Soubor  Úpravy  Zobrazení  Komunikace  Akce  Okno  nápověda

File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help

EDIT      /usr/lpp/netdata/macroisz/PLKZSPSP.Z      Columns 00001 00072
Command ==>      Scroll ==> HALF

000106  @DTW_ASSIGN(CASTKA, UCASTKA1)
000107  @DTW_INSERT(CASTKA, " ", CASTKA)
000108  @DTW_INSERT(CASTKA, UCASTKA2, CASTKA)
000109  @DTW_INSERT(CASTKA, " ", CASTKA)
000110  @DTW_INSERT(CASTKA, UCASTKA3, CASTKA)
000111  %ENDIF
000112  @DTW_SUBSTR(UOSLOV, "1", "1", UPOHL)
000113  %IF ($ (UPOHL) == "M")
000114      @DTW_ASSIGN(TEXT_OSLOV1, "pan")
000115      @DTW_ASSIGN(TEXT_OSLOV2, "narozen")
000116  %ELSE
000117      @DTW_ASSIGN(TEXT_OSLOV1, "paní")
000118      @DTW_ASSIGN(TEXT_OSLOV2, "narozena")
000119  %ENDIF
000120  <BODY bgcolor="#ffffff" leftmargin= "50">
000121  <font size="4">
000122  <BR>
000123  <DIV align="right">
000124      %IF ($ (UCSMLOUVY) == "")
000125          Smlouva číslo:_____

04/015
  
```

[Návazné spojení se vzdáleným serverem/hostitelem mvrsl s použitím logické jednotky/společné oblasti X900v/C46 a portu 23]
 [RICOH Alcio MP C3300 PCL 6 R426 na Sale0002]

Obrázek 8: Makro v prostředí TSO

8.2 Migrace

Vzhledem k výše řečenému, sestavy v prostředí TSO nemají export jako takový, jsou ukládána na serveru pouze binárně. Každá operace, změna i samotná práce s takto interpretovaným jazykem, který je možno zobrazit pouze v TSO prostředí, je málo komfortní.

Prvním krokem je získání zdrojového kódu sestavy. V prostředí TSO jsou na výběr pouze dvě možnosti. Jednou je doslovný přepis kódu, druhou je kopírování po čtyřřádkové řádkách. Bohužel i takto kopírovaný text se musí dále editovat (kopírované řádky obsahují číslování řádku, atp.). Jak jsme již uvedli, použijeme HTML část kódu. Můžeme si pomoci textového editoru ulehčit práci, pokud umí zvýraznit syntaxi programovacích jazyků. Na obrázku 9 můžeme vidět, jak to vypadá v editoru Notepad ++. Barevně odlišený kód je HTML, zbytek je funkcionální jazyka Net Data.

```

1  %HTML( INPUT) {
2  @DTW_DATE("S",DATE)
3  @DTW_SUBSTR(DATE,"7","2",DEN)
4  @DTW_SUBSTR(DATE,"5","2",MES)
5  @DTW_SUBSTR(DATE,"1","4",ROK)
6  <FORM NAME=VSTUP METHOD=POST ACTION=REPORT TARGET=NOVE_OKNO>
7  <H2>Smlouva o poskytnutí sociálního příspěvku</H2>
8  <HR></HR>
9  <table border="0" with=100%>
10 <tr>
11 <td width=40%>akciová společnost</td>
12 <td width=60%>
13 <SELECT size="1" name="UAKC">
14 <OPTION value="0">ISPAT Nová Huť,a.s.</OPTION>
15 <OPTION value="12" selected>Vysoké Pece Ostrava, a.s.</OPTION>
16 </SELECT>
17 </td>
18 </tr>
19 <tr>
20 <td width=40%> číslo zaměstnance</td>
21 <td width=60%>
22 <input type="text" name="UCPRAC" size="6" length="6"></td>
23 </tr>
24 </table>

```

Hyper Text Markup Language file length: 1858 lines: 24 Ln: 24 Col: 69 Sel: 0

Obrázek 9: Práce v Notepad++

Nyní přistoupíme k nahrazení funkci a logiky programu z Net Data. Založíme JSP v prostředí WSAD 5.0. Rozdíly v implementaci bych uvedl na pár příkladech

8.2.1 NAHRAZENÍ LOGICKÉHO ROZHODOVÁNÍ

Net Data

```
%IF ($ (UCSMLLOUVY)=="")
Smlouva cislo _____
%ELSE
Smlouva cislo : <B>$(UCSMLLOUVY)</B>
%ENDIF
```

Výpis 3: Funkce v NET Data

JSP(JSP Out)

```
<%!private String  cissmlouvy;%>
<% cissmlouvy = (String)session.getAttribute("UCISSMLLOUVY");
    if(cissmlouvy.equals("")){cissmlouvy = "_____";}
```

Výpis 4: Funkce v JSP

8.2.2 PRÁCE SE SYSTÉMOVÝMI KNIHOVNAMI.

Net Data

```
@DTW_DATE("S",DATE)
@DTW_SUBSTR (DATE,"7","2",DEN)
@DTW_SUBSTR (DATE,"5","2",MES)
@DTW_SUBSTR (DATE,"1","4",ROK)
```

Výpis 4: Práce s daty v jazyce NET Data

První řádek obsahuje funkci *DATE*, která nám nastaví formát data a výstup řetězce (@DTW_DATE(format, stringOut)). Na výběr je mnoho formátů, uvedu například tyto tři:

- Dny v roce (1 - 366), pomocí písmene D,
- evropský (dd/mm/yy), pomocí písmene E,
- standardní (yyyymmdd), pomocí písmene S.

Písmenem S jsme zvolili standardní formát data a ukládáme jej do proměnné DATE. V dalších řádcích vybíráme části řetězce pomocí funkce *SUBSTR* (@DTW_SUBSTR(stringIn, n, length, pad, stringOut)) tak, aby odpovídali skutečnosti.

Java

V JSP použijeme volání java třídy *CplkDatum*, který již obsahuje jednotlivé metody práce s daty. V tomto případě použijeme metodu *getSysDatum*, která nám vrátí systémové datum.

```
<jsp:useBean id="datum" class="cz.novahut.plx.CplkDatum"></jsp:useBean>

public String getSysDatum(){
    java.util.Date dnes = new java.util.Date();
    SimpleDateFormat formatter=new SimpleDateFormat("dd.MM.yyyy");
    try {
        dateOut = formatter.format(dnes);
        mesic = dateOut.substring(3,5);
        rok = dateOut.substring(6);
        den = dateOut.substring(1,2);
        return dateOut;
    } catch (IllegalArgumentException iae) {
        System.out.println("Error:sysDate " + iae.getMessage());
        return null;
    }
}
```

Výpis 5: Práce s daty v jazyce Java

8.2.3 PŘEDÁVÁNÍ ŘÍZENÍ Z JSP DO SERVLETU

Po vyplnění údajů ve vstupním JSP a potvrzení tlačítkem dojde k předání parametru do servletu.

```
<FORM NAME=Form1 METHOD="POST" target="_blank"
action="../servlet/servlets.Splkpersonaliste" onSubmit="return vetIn();">
```

Výpis 6: Volání metody POST

Již v samotném servletu je volána funkce *doPost*, a po následně identifikaci bude zavolaná příslušná třída pro zpracování.

```
public void doPost(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
```

Výpis 7: Volání metody doPOST

Inicializace třídy

```
CperSmlSocPrisp smlsocprisp = new CperSmlSocPrisp();
```

Výpis 8: Inicializace třídy CperSmlSocPrisp

Zpracování a identifikace aplikace

V servletu dojde k identifikaci aplikace a následné nastavení vstupních parametrů ve třídě CperSmlSocPrisp pomocí metody set

```
if(aplikace.equals("SMLSOCPRIS")){
    smlsocprisp.setAkc(Integer.valueOf(req.getParameter("UAKC")).intValue());
    smlsocprisp.setCprac(Integer.valueOf(req.getParameter("UCPRAC")).intValue());
    smlsocprisp.setDatVyb(req.getParameter("UDATUM"));
    smlsocprisp.setCastka(Integer.valueOf(req.getParameter("UCASTKA")).intValue());
}
```

Výpis 9: Zpracování parametrů

Zpracování dat v CperSmlSocPrisp třídě s použitím SQLJ

```
smlsocprisp.Zpracovani();
```

Výpis 10: Zpracování dat s použitím SQLJ

Předání parametru výstupnímu JSP (per_SmlouvaSocPrisp_out.jsp)

```

session.setAttribute("UMESVYPL",new
String(req.getParameter("UMESVYPL").getBytes("ISO-8859-1"),"iso-8859-2"));

session.setAttribute("UCISSMLOVY",String.valueOf(req.getParameter("UCISSMLOVY")));

session.setAttribute("UDATUM",String.valueOf(req.getParameter("UDATUM")));

resultPage = "/protected/per_SmlouvaSocPrisp_out.jsp";

req.setAttribute("smlsocprisp",smlsocprisp);

```

Výpis 11: Předání parametru

Zpracovávání dat

V Net Data bylo použito volání uložené procedury.

```

%FUNCTION(DTW_SQL) VYBER1 (
IN DECIMAL(3) UAKC, IN DECIMAL(7) UPRAC, IN INTEGER UCASTKA,
OUT CHAR(20) UJMENO, OUT CHAR(20) UPRIJM, OUT CHAR(10) UTITUL1,
OUT CHAR(10) UTITUL2, OUT CHAR(10) UDATNAR, OUT CHAR(4) UOSLOV,
OUT CHAR(122) UBYDLISTE, OUT CHAR(402) UTEXT,
OUT INTEGER USQLCOD, OUT CHAR(5) USQLSTAT, OUT CHAR(1) UERROR,
OUT CHAR(78) UODEZVA, OUT CHAR(20) UPROCES, OUT CHAR(10) UAPPL) {
CALL UPLKPSP

```

Výpis 12: Volání uložené procedury v jazyce NET Data

V tomto případě už jde o výběr osobních údajů jednoho zaměstnance, proto se využilo již dříve naprogramovaných SQLJ, které nám tyto údaje poskytnout.

V Java třídě *CperSmlSocPrisp* na výpisu 13, ve kterém si vytvoříme připojení na SQLJ, pro předávání vstupních a výstupních dat. Např. Pro získání jména a příjmení použijeme SQLJ *JplKD02*, která z databáze vrátí jméno a příjmení na základě čísla zaměstnance.

```

JplKD02 kd02 = new JplKD02();

```

Výpis 13: Připojení pomocí SQLJ

Pro zpracování vstupních a výstupních parametrů, použijeme metody set a get. Příklad pro získání informací podle identifikátoru, v našem případě číslo zaměstnance.

```
public void setCprac(int cprac){this.cprac = cprac;}
```

Výpis 14: Zpracování vstupního parametru čísla zaměstnance

Ke zpracování dojde v metodě VyberKD02

```
public void VyberKD02() throws Exception{ //zavolání metody
try{
kd02.setCon(conectDB.getCon()); //nastavení vstupních parametrů
kd02.setAkc(akc);
kd02.setCprac(cprac);
kd02.setDatVyb(datVyb);
kd02.ctiKD02(); //samotný sql výběr
switch(kd02.getKd02Pohl().charAt(0)){ //zpracování výběru
case 'M' : osloveni = "pan"; break;
case 'Z' : osloveni = "paní"; break;
}
}
```

Poté vytvoříme řetězec, který odešleme zpět pomocí JSP out.

```
jmenoPrijmeni = kd02.getKd02Titul1() + kd02.getKd02Jmeno() +
kd02.getKd02Prijmeni() + kd02.getKd02Titul2(); //nastavení proměnných
pro výstup
}catch (SQLException e) { //zachytávání vyjímek
odezva = kd02.getOdezva();
throw e; }
```

Výpis 15: Zpracování dat pomocí SQLJ

Nastavení výstupu z java třídy pomocí metody get. Například kd02 jméno a příjmení.

```
public String getJmeno(){return jmenoPrijmeni;}
```

Výpis 16: Práce s get parametrem

V JSP out zobrazeno jméno a příjmení tímto způsobem

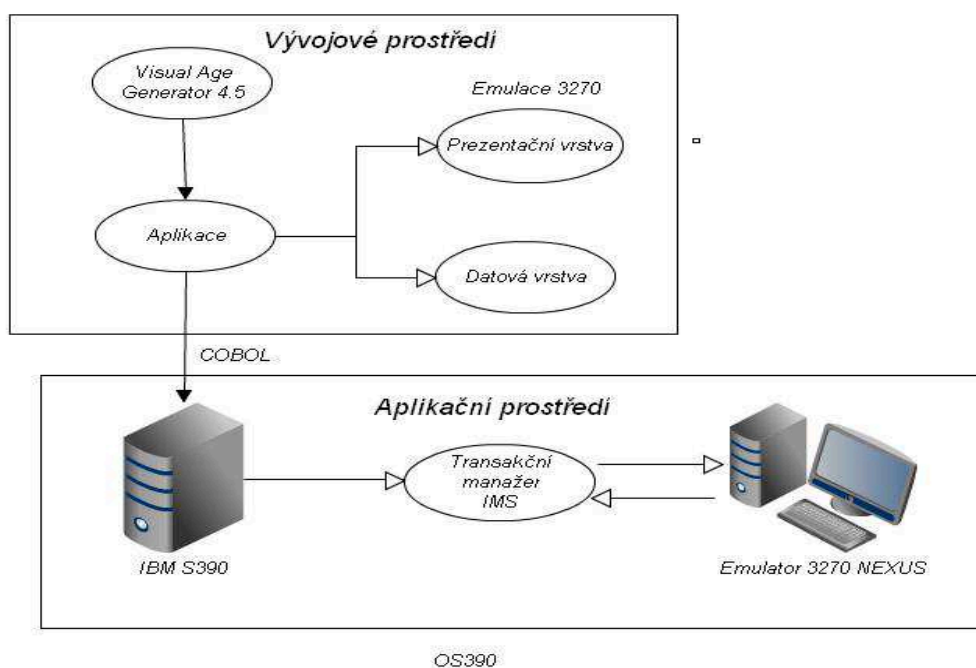
```
<%=smlsocprisp.getJmeno()%>
```

Výpis 17: Zobrazení jména a příjmení v JSP out

9. Migrace z transakčního prostředí do webového za využití uložených procedur DB2

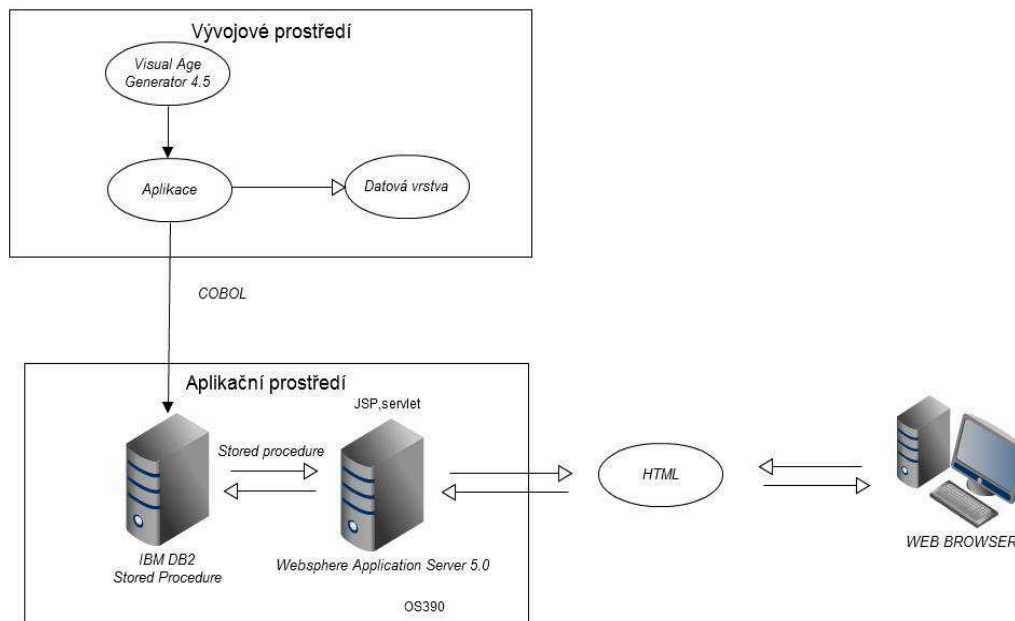
V současné době hlavní metoda zpracování a editace dat probíhá pomocí transakčního manažeru IMS, kdy přes terminály 3270 uživatelé zadávají a editují personální a mzdová data v systému. Z licenčního důvodu se připravuje ukončení provozu transakčního manažeru, což ve výsledku znamená migraci aplikaci do webového prostředí. Aby nebylo nutné zdlouhavému nahrazování programu novými, je snaha využít stávajícího kódu pro výběr a zpracování dat pomocí generovaných uložených procedur DB2. Nahrazení prezentační vrstvy 3270 pomocí JSP.

9.1 Proces s využitím transakčního manažeru



Obrázek 10: Předchozí verze

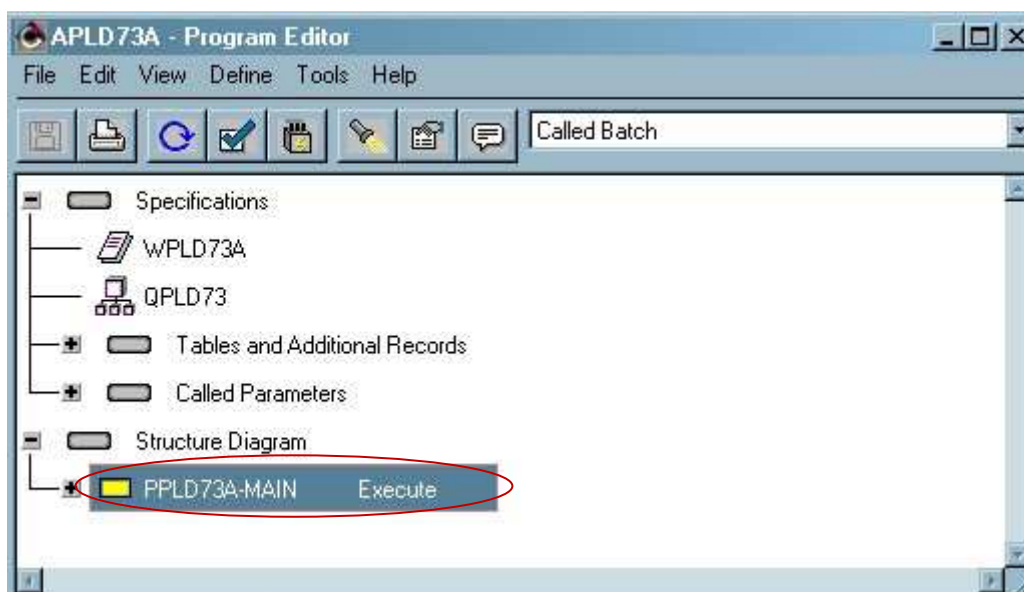
9.2 Proces s využitím webových služeb



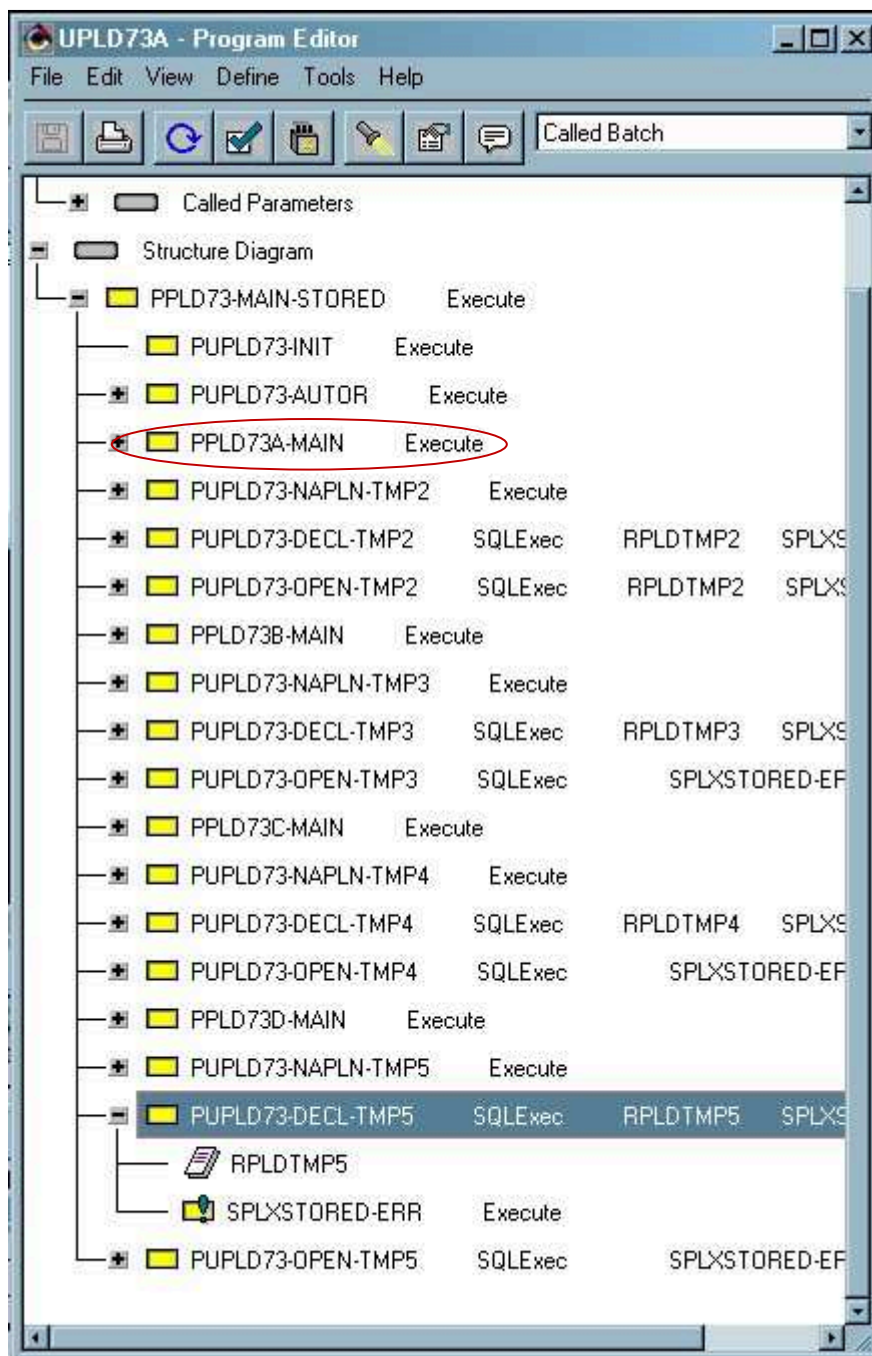
Obrázek 11: Současná verze

9.3 Vytvoření uložené procedury DB2

Ve vývojovém prostředí Visual Age Generator, kde jsou vyvíjeny aplikace pro transakční zpracování, kdy část aplikace tvoří prezentační vrstvu (main transaction), která využívá aplikace na business vrstvě, jinými slovy called batch. Z této called batch využijeme hlavní proces PLD73A-MAIN s SQL příkazy pro zpracování dat, který je společný i pro zpracování dat v uložené proceduře, viz obrázek 12



Obrázek 12: Called batch v prostředí VAGEN

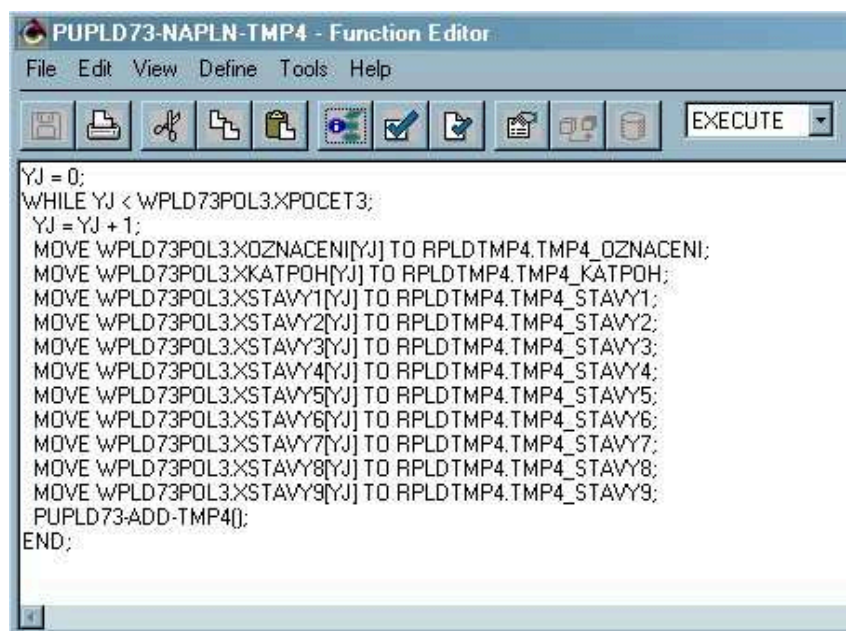


Obrázek 13: Znovu použita jako uložená procedura.

Na obrázku 13 vidíme použití stejné funkce PLD73A-MAIN v novém programu, který bude později spouštěn jako uložená procedura.

Naplnění temporary table

V uložené proceduře řešíme pouze jiné předání dat do prezentační vrstvy, daty ze společné pracovní oblasti naplníme do dočasné tabulky (temporary table). Její obsah se následně pomocí kurzoru předává k dalšímu zpracování v java třídě. Oproti zpracování transakčním managerem jsou data předána k řízení do prezentační vrstvy.



Obrázek 14: Naplnění temporary table

Deklarace a otevření kurzoru

Temporary table není předávána jako parametr, ale formou otevřeného kurzoru, který bude uzavřen až po zpracování dat v java třídě.

```
DECLARE CPLDTMP2 CURSOR WITH RETURN FOR SELECT * FROM
NHPLDCPD.PLDG_TMP2 //deklarace kurzoru

OPEN CPLDTMP2 //otevření kurzoru
```

Výpis 18: Práce s kurzorem

9.3.1 ZPRACOVÁNÍ V JAVA TŘÍDĚ

Základní deklarace

Definice java metod pro práci s uloženými procedurami.

```
CallableStatement cstmt;
PreparedStatement stmt;
```

Výpis 19: Deklarace metod

Deklarace result setu

```
ResultSet rs = null;
```

Výpis 20: Deklarace result setu

Deklarace java objektu vektor

```
Object[] row;
Vector VectorTAB = new Vector();
```

Výpis 21: Deklarace vektoru

Samotné volání uložené procedury DB2 v Java třídě

```
public void callStoredProcUPLD73A() throws SQLException {
    cstmt = con.prepareCall("call
    UPLD73A(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
}
```

Výpis 22: Volání uložené procedury

Zpracování vstupních a výstupních parametrů procedury

```
cstmt.setString(1, super.getUtypeses());
cstmt.setInt (2, super.getUakcn
cstmt.setString(3, super.getUzav());
cstmt.setString(4, super.getUprovoz());
cstmt.setString(5, super.getUnstrod());
cstmt.setString(6, super.getUnstrdo());
...

```

Výpis 22: Zpracování parametrů

Samotná data z temporary table jsou předány do result setu a data jsou uložena do java objektu vector.

```
rs=cstmt.executeQuery();
```

Výpis 23: Práce s result setem

Předání dat z result setu do java objektu vektor

Ve smyčce vybíráme data z result setu do vektoru.

```
Object radek[] = new Object[14];
VectorTAB.addElement(radek);
radek[1] = rs.getString("TMP2_OZNACENI"); //plnění vektoru
...

```

Výpis 24: Předání dat

Předání objektu vektor proběhne metodou GET

```
public Vector getVectorTAB() {return VectorTAB;}
```

Výpis 25: Předání objektu

Práce s vektorem ve výstupním JSP

V tomto případě je výstupem tabulka, jejichž jednotlivé řádky budou generovány řádky z objektu vektor

```
<TR class="barva_def" onmouseover="this.className='barva_zvyraz'"
onmouseout="this.className='barva_def'">

<%for(i=1;i<pocetSloupcu;i++); // základní smyčka pro výběr řádku
vektoru

if(String.valueOf(row[0]).equals("H")){ //TISK HLAVIČKY

<TH bgcolor="#334299" valign="top" align="center" >

<FONT color="#ffffff"> <%= String.valueOf(row[i])%></FONT></TH>

<%}if(String.valueOf(row[0]).equals("R")){ //TISK ŘÁDKŮ
if(String.valueOf(row[1]).equals("SUMA")){//TISK ŘÁDKŮ%>

<TD align="right" bgcolor="#C8CEDA"><%= String.valueOf(row[i])%></TD>

<%}else{ %>

<TD align="right"><%= String.valueOf(row[i])%></TD>
```

Výpis 26: Vektor v JSP out

10. Závěr

Výsledkem je intranetový portál, který je v současné době plně využíván. Výhodou pro běžné uživatele je rozhodně vizualizace pomocí HTML, která zvyšuje komfort práce a interaktivitu prostředí, s kterým dennodenně pracují. Další kladnou vlastností je správa portálu. Oproti předchozí verzi je tvořen jedinou aplikací, která pomocí Websphere Application Server 5.0 je velice jednoduše a efektivně spravována.

Tato práce pro mě byla velkým přínosem, co se nových zkušeností týče. Vyzkoušel jsem si práci v nadnárodní korporaci, sdílel své názory s týmem lidí. Také jsem pracoval se zajímavými produkty firmy IBM, se kterými jsem se dříve nesetkal.

Z hlediska dalšího vývoje výsledné práce je pravděpodobná příprava na migraci do prostředí Websphere Application Server verze 7.

11. Reference

- [1] ArcelorMittal a.s. [online]. 2012 [cit. 2012-04-22] ArcelorMittal a.s. Dostupné z WWW: < <http://arcelormittal.cz/>>
- [2] IBM [online] 2012 [cit. 2012-04-22] IBM Dostupné z WWW :< http://www-03.ibm.com/systems/resources/systems_i_software_netdata_db2rn.pdf>
- [3] STANÍČEK, Petr. *CSS Kaskádové styly: kompletní průvodce*. 2. vyd. Brno: Computer Press, 2003, 178 s. ISBN 80-722-6872-4.
- [4] ŠKULTÉTY, Rastislav. *JavaScript: programujeme internetové aplikace*. Vyd. 1. Praha: Computer Press, 2004, 224 s. ISBN 80-251-0144-4.
- [5] BOLLINGER, G. *JSP Java, Server, Pages*. Vyd. 1. Praha: Grada, 2003, 418 s. ISBN 80-247-0340-8.
- [6] IBM [online] 2012 [cit. 2012-04-22] IBM Dostupné z WWW :< <http://www-01.ibm.com/software/webservers/appserv/was/> >
- [7] Redbooks IBM [online] 2012 [cit. 2012-04-22] IBM Dostupné z WWW :< <http://www.redbooks.ibm.com/redbooks/pdfs/sg246503.pdf> >